

New formats and interfaces for multi-document news summarization and its evaluation

Bettina Berendt¹, Mark Last², Ilija Subašić¹, Mathias Verbeke¹

¹ *KU Leuven, Belgium*

² *Ben-Gurion University of the Negev, Israel*

Last author draft. To appear in A. Fiori (Ed.), *Innovative Document Summarization Techniques: Revolutionizing Knowledge Understanding*, IGI Global, 2013.

ABSTRACT

News production, delivery, and consumption are increasing in ubiquity and speed, spreading over more software and hardware platforms, in particular mobile devices. This has led to an increasing interest in automated methods for multi-document summarization. We start this chapter with discussing several new alternatives for automated news summarization, with a particular focus on temporal text mining, graph-based methods, and graphical interfaces. Then we present automated and user-centric frameworks for cross-evaluating summarization methods that output different summary formats, and describe the challenges associated with each evaluation framework. Based on the results of our user studies, we argue that it is crucial for effective summarization to integrate the user into sense-making through usable, entertaining and ultimately useful interactive summarization-plus-document-search interfaces. In particular, graph-based methods and interfaces may be a better preparation for people to concentrate on what is essential in a collection of texts, and thus may be a key to enhancing the summary evaluation process by replacing the “one gold standard fits all” approach with carefully designed user studies built upon a variety of summary representation formats.

1 INTRODUCTION

Automatic text summarization is a computer-aided process of distilling the most important information from one or several textual sources on a common topic. Based on the number of input sources, summarization methods are categorised into single-document and multi-document approaches. The output of most summarization methods is a natural-language text in itself.

One distinguishes between summarization by extraction and by abstraction (Hahn & Mani, 2000).

Extractive methods select elements from the original text(s) and recompile them to form the summary.

Abstractive methods build an internal semantic representation and then use natural-language generation techniques to create a summary. Due to the remaining limitations of natural-language generation methods, extractive summarization remains the dominant approach today.

In this chapter, we focus on extractive summarization of text streams, particularly news streams. In this domain, *multi-document* summarization is especially relevant because of the multitude of sources.

Popular applications of these ideas are news aggregators such as Google Newsⁱ or Yahoo Newsⁱⁱ. However, these do not aim at producing a coherently-seeming natural-language text, but a search-engine-type collation of information. This echoes the findings of Barzilay, McKeown, and Elhadad (1999) that pure extraction may be inappropriate in multi-document summarization especially of news texts, because it may produce summaries which are overly verbose or biased towards some sources. In contrast to the static methods of multi-document summarization, STORIES (Subašić & Berendt, 2010a) is designed for both summarization and search in a time-indexed collection of documents. STORIES uses *story graphs* to represent summaries.

The classical summary evaluation methods suffer from severe limitations, especially when applied to Temporal Text Mining techniques, which may produce summaries in both textual and graph formats. Hence, we present an automated and a user-centric framework for the cross-evaluation of news summaries. An initial user study of the proposed frameworks provides encouraging results. We use these results to outline directions for future work.

The contribution of this chapter is threefold: *First*, we give an overview of key concepts and formats of text summarization in general and news summarization in particular (Section 2). *Second*, we present a critical re-assessment of evaluation methodology. In Section 3, we discuss the issue of evaluation of summarization methods, highlighting in particular the difficulties of cross-evaluating text-based as well as the emerging multitude of non-text-based summaries. We also take a closer look at a form of human input that is crucial for all automated evaluations: the “ground truth” summaries and their construction.

We argue that current practice severely limits our ability to evaluate automatic methods and that this calls for interactive, graphical approaches. *Third*, in Sections 4 and 5 we propose a new approach to evaluating summarization methods. We describe the automated and the user-centric frameworks for cross-evaluating methods that output different summary formats, and we describe the challenges associated with each evaluation framework. In Section 6, we draw conclusions from these findings and outline key elements of future multi-document summarization methods.

2 KEY CONCEPTS AND FORMATS IN TEXT SUMMARIZATION

Extractive summarization aims at the selection of a subset of the most relevant fragments from a source text into the summary. The fragments can be paragraphs (Salton, Singhal, Mitra, & Buckley, 1997), sentences (Luhn, 1958), keyphrases (Turney, 2000; Litvak, Aizenman, Gobits, Last, & Kandel, 2011) or keywords (Litvak & Last, 2008). Extractive summarization usually consists of *ranking*, where each fragment of a summarized text gets a relevance score, and *extraction*, where the top-ranked fragments are gathered into a summary, according to their appearance in the original text.

While in *generic summarization*, the only input for a system is a document (or documents) to summarize, in *query-based summarization* a query expressing the user’s interest has to be provided. In a query-based summary, the fragment rankings must take into account the content of a given query, making the extract adaptable to the user’s information needs.

Various text representation models have been utilised across different summarization approaches. In addition to the traditional “bag-of-words” model, graphs are becoming increasingly popular, due to their ability to enrich the document representation with syntactic and semantic relations. Erkan and Radev (2004) and Mihalcea (2005) introduced LexRank and TextRank, respectively – algorithms for unsupervised extractive summarization that rely on the application of iterative graph-based ranking algorithms, such as PageRank (Brin & Page, 1998) and HITS (Kleinberg, 1999). Their methods represent a document as a graph of sentences interconnected by similarity relations. MUSE (Litvak & Last, 2012) uses two graph-based models, which are based on sentence and word segmentation, respectively. MUSE also uses features calculated from the document structure and its vector-space representation.

In addition to these questions of format, two features of the underlying corpora have received increased attention in recent years: multilinguality and temporality (for example, see the multilingual multi-document summarization task and the multilingual summary evaluation tracks at MultiLing 2013ⁱⁱⁱ). We

will present key features of multilingual summarization with reference to the MUSE approach, then give an overview of approaches in temporal summarization with a focus on representation formats, and illustrate this with the STORIES approach.

2.1 The MUSE approach for multilingual static summarization

The MUSE (MULTilingual Sentence Extractor) approach (Litvak & Last, 2012) uses a linear combination of 31 language-independent features from various categories for ranking each sentence in a document. Language-independent features do not require any morphological or syntactic analysis of the summarized text, and they may include the sentence position in a document, the number of characters and words in a sentence, the similarity of the sentence to the document title, and other statistical metrics. In contrast, language-specific features of a sentence may include the presence of cue words or the number of certain parts-of-speech. For example, “to sum up” in English may be an indicator that the following is a summarizing sentence, and a high number of nouns in a sentence may indicate a high amount of topical information in this sentence. MUSE finds the best set of feature weights by a genetic algorithm trained on a collection of human-generated document summaries. Obviously, the quality of the resulting sentence ranking model directly depends on the number and the quality of the training summaries. Formally, the MUSE model for sentence scoring can be expressed by the following formula:

$$Score = \sum w_i \times r_i,$$

where r_i is the value of i^{th} sentence feature and w_i is its weight in the linear combination.

The MUSE approach builds upon two main modules: the *training module* activated offline, and the *summarization module* operating online.

The *training module* receives as input a corpus of documents, each accompanied by one or several gold-standard summaries: abstracts or extracts compiled by human assessors. The set of documents may be either monolingual or multilingual, and their summaries have to be in the same language as the original text. As a second parameter, the module obtains a user-specified set of sentence features computed by the system. Then, the training module applies a genetic algorithm to sentence-feature matrices of pre-computed sentence scores for each input feature, with the purpose of finding the best linear combination of features that maximises the value of ROUGE (Lin & Hovy, 2003, see Section 3.1 below) as a fitness function. The output of the training module is a vector of weights for user-specified sentence ranking features.

The *summarization module* performs an on-line summarization of input text(s). Each sentence of an input text document obtains a relevance score according to the trained model, and the top-ranked sentences are extracted to form the summary in their original order. To avoid duplicate content, a new sentence is added if and only if it is not similar to the previously selected sentences. The length of the resulting summaries is limited by a user-specified value (maximum number of words or sentences in the text extract or a maximum extract-to-text ratio). The summarization module is expected to use the model trained on the same language as that of the input texts. If such a model is not available (no annotated corpus in the text language), the user can choose from the following: (1) a model trained on some other language/corpus, or (2) a model trained on a parallel corpus generated by a machine translation tool. Both approaches were shown by Litvak and Last (2012) to provide a reasonable performance while saving significant annotation efforts.

2.2 Summarizing temporally-indexed texts

The above methods are not directly applicable to the news summarization domain, as they ignore the potentially dynamic nature of the summarized texts. Moreover, they all produce output summaries in strictly textual, non-interactive format, which may not represent well the temporal characteristics of a

news stream. Temporality is, on the one hand, a feature of the documents in a news corpus. On the other hand, it may govern what is expected of a summary. In particular, readers often want to know what happened in a particular time period, such that the expected summaries are query-based, with the query being a time period. Further restrictions may be added, as we discuss with respect to the STORIES method below. We begin with an overview of prominent approaches to news summarizations in the literature.

First Story Detection (FSD; Allan, Lavrenko, & Jin, 2000) is a Topic Detection and Tracking (TDT; Allan, 2002) task incorporating the temporal dimension of a corpus. Sometimes this task is also referred to as the Online News Event Detection (ONED) task. In this task, a system has to decide whether a newly arrived document discusses “a new event” or not. In TDT, events are defined as “something that happens at a particular time and place” (Allan, Lavrenko, & Jin, 2000). During the seven years in which the TDT framework was active, a number of approaches to solving the FSD task have been developed. Allan, Lavrenko, and Jin (2000) generated queries and used cosine similarity to compare them with already seen documents. Luo, Tang and Yu (2007) tackled these problems and explored document source quality, efficient indexing, and user interfaces for FSD. An interesting extension of the FSD task was described by Nallapati, Feng, Peng, and Allan (2004): going beyond detecting events towards discovering relations between events. A more recent application of FSD to social-network status update data was described by Petrović, Osborne, and Lavrenko (2010).

The DUC (Document Understanding Conference) Update Summarization task^{iv v} outputs a set 100-word summary for a number of time periods so that the summaries contain information from previous periods. A LexRank-inspired approach to update summarization was proposed by Chen and Chen (2008). Other methods specifically designed for update summarization used information distance-based summaries (Long, Huang, Zhu, & Li, 2009), integer linear programming (Gillick & Favre, 2009), or latent semantic analysis (Steinberger & Ježek, 2009).

In the TREC Novelty Detection Task, participants are given a set of documents on a topic, and their task is to extract sentences that are relevant to the topic and to select sentences that are “new”. In this context, “new” is defined as containing information that has not appeared previously in a topic’s set of documents. Experiences with running the TREC Novelty Detection Task were summarized by Soboroff and Harman (2005). In 2012, TREC started the Temporal Summarization task.^{vi}

In contrast to the previous tasks that focus on the detection of a new element or breakpoint in a stream of documents, *story tracking* is the activity of following one story over time and summarizing it. This task has been popularized by Temporal Text Mining. *Temporal Text Mining (TTM)* methods output bursty (novel) patterns that point to the changes in the story they track, and the subjects arising from these changes. *Subjects* constitute the high-level story; they can be events (e.g. a specific ski slalom in the Winter Olympics) or topics (e.g. doping).^{vii} The patterns consist of *story elements*, syntactical units extracted from the underlying documents. For example, an element could be a term, and the pattern could be this term plus some score assigned to it. We also define a *story representation* as a set of bursty story elements used to represent a subject. Story elements have different levels of expressiveness. TTM methods operate on sub-sentence story elements, and we distinguish the following *elements*: tokens, n-grams, and n-gram groups. Further filters are possible, producing elements such as terms with frequencies or other weights above a threshold, or n-grams denoting named entities.

A *token* is a series of characters not containing any of a set of predefined delimiters. *N-grams* are content-bearing tokens. Basic n-grams are unigrams (1-grams), where every token is a unigram. More advanced n-grams are sequences of n contiguous (or non-contiguous) tokens extracted from the text. Non-consecutive, or skip- m n-grams, contain n tokens appearing in a window of m tokens. *N-gram groups* are collections of n-grams pointing to the same subject. These groups can be n-gram cluster centre values, latent variables’ probability distributions over n-grams, or some other way of grouping by similarity. In general, elements are associated with some weight. We focus on the *burst scores* assigned by the respective TTM method. This score may be a probability of an element appearing in a bursty subject, the relative importance in a bursty subject cluster centre, or a weight in a latent component. Regardless of

their mathematical specifics, these burst scores provide an ordering on the elements that can be used for ranking the elements in query generation.

Based on the differences in their story representations, we distinguish between three types of TTM tracking methods: (a) keyword representation, (b) group representation, and (c) combo representation methods. Type (a) methods use a list of bursty n-grams ranked by their burst scores (Kleinberg, 2002; Fung, Yu, Yu, & Lu, 2005; Gruhl, Guha, Kumar, Novak, & Tomkins, 2005; He, Chang, Lim, & Zhang, 2007; Smith, 2002). Type (b) methods assemble bursty n-grams into groups which point to subjects (Fung et al., 2005; Wang & McCallum, 2006; Mei & Zhai, 2005; Schult & Spiliopoulou, 2006; Janssens, Glänzel, & Moor, 2007). Type (c) methods use a combination of the previous two approaches (Subašić & Berendt, 2010a; Allan, Gupta, & Khandelwal, 2001).

2.3 The STORIES approach for temporal summarization

STORIES (Subašić & Berendt, 2010a) comprises summarization and search. Both steps are time-indexed and therefore user-directed in the STORIES tool (Subašić & Berendt, 2010b). For pre-processing, a corpus of text-only documents is transformed into a sequence-of-terms representation. Basic term statistics are calculated to identify candidates for story basics: the top-ranked words based on term frequency, TF.IDF weights, regular terms plus named entities, and all terms.

For the summarization of a time-indexed subset of the whole corpus, c_t for period t , the frequency of the co-occurrence of all pairs of content-bearing terms b_j in documents is calculated as the number of occurrences of both terms in a window of w terms, divided by the number of all documents in c_t . We call this measure *local relevance* with $LR_t(b_1, b_2) = freq_t(b_1, b_2)$. LR normalised by its counterpart in the whole corpus C yields *time relevance* as the measure of burstiness: $TR_t(b_1, b_2) = (freq_t(b_1, b_2) / freq_C(b_1, b_2))$. Thresholds are applied to avoid singular associations in small sub-corpora and to concentrate on those associations that are most characteristic of the period and most distinctive relative to others. This gives rise to the *story graphs* $G_t = \langle V_t, E_t \rangle$. The edges E_t are the story elements: all pairs (b_1, b_2) with absolute frequencies and TR above the respective thresholds. The nodes are the terms involved in at least one association in this symmetric graph: $V_t = \{ b_j / \exists b_k : (b_j, b_k) \in E_t \}$. From each document, we extract sentences containing “facts”, short statements with semantic role labelling, as returned by Open Calais^{viii}. The full set of these sentences for each time period is indexed using Lucene^{ix}. We then use story graphs to filter the most important facts: for each of the graph’s edges, we query the index, using node names of the edge as query terms, and select the top sentences as defined by Lucene. We treat the resulting set of short textual statements as a summary of the story.

Search can be constrained by the nodes of a subgraph of the story graph. Retrieval is then restricted to documents relevant to these subgraphs. The selection of documents of the starting corpus C corresponds to a top-level query; this query is expanded by all nodes of the subgraph as additional query terms, subject to the time restriction.

Thus, like MUSE, STORIES focusses on graph representations. In contrast to MUSE graphs, which represent a single text document, STORIES graphs represent a multi-document story. Another difference is that in the MUSE graph representation, the window size w is usually equal to one (the co-occurring terms have to be adjacent to each other). Both approaches extend the bag-of-words idea by taking co-occurrence as well as sequence into account. Other summarization approaches extend this by forming a graph based on additional syntactic and/or semantic information. For example, based on semantic role labelling and the merging of synonyms, triples can be formed and event templates filled (Trampus & Mladenic, 2009).

3 CHALLENGES IN EVALUATING AUTOMATED TEXT-BASED SUMMARIES

In this section, we first give a brief overview of classical cross-evaluation methods and sketch their limitations. We then present a cognitive hypothesis that allows us to make different formats of summaries comparable.

3.1 Classical evaluation methods of textual summaries

Summary quality can be evaluated in an *intrinsic* or *extrinsic* manner, depending on whether the goal is to assess the summary itself (i.e. according to its information content) or its effectiveness for reaching the goals of a third-party application (e.g. information retrieval). In intrinsic evaluation, the summary informativeness is determined by comparing its content to a gold standard summary. In the case of the extrinsic evaluation, the third-party system performance is evaluated. For example in the text categorisation domain, a document summary can serve as a noise filter, and the question that should be asked is whether categorization performance can be increased by using the summary rather than the entire document.

Another aspect for summary evaluation approaches is the automation issue. Evaluation can be human-assisted or automated. In human-assisted evaluation approaches, human participation is an integral part of the evaluation process. For example, in Pyramid evaluation (Nenkova, Passonneau, & McKeown, 2007), pieces of information that are conceptually equal across model summaries (Summary Content Units) are manually identified. In the DUC'04 conference^x, summaries were evaluated with respect to linguistic quality aspects (grammaticality, non-redundancy, referential clarity, focus, structure, and coherence). Each quality aspect has a related question to be answered by human assessors according to a five-point scale. In automated evaluation approaches, a summary quality is determined exclusively by computers. For example, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin & Hovy, 2003) automatically compares a computer-generated summary to gold standard summaries (generated by human annotators) in terms of overlapping units such as n-grams, word sequences, and word pairs. It has been shown that a quality ranking of summaries by the ROUGE measures is highly correlated with a quality ranking of these summaries by humans (Lin & Hovy, 2003), meaning that ROUGE scores can select the “best” summaries.

3.2. The problem of the “ground truth” summaries

Automated evaluation methods use text similarity measures (such as different variations of ROUGE) to compare system summaries to human (peer) summaries provided by native language speakers (assessors). Usually, the assessors are required to spend a certain amount of time on reading each article from a conventional computer screen and then choose the most important (key) sentences containing a pre-defined number of words. One popular collection of summarized documents in English is the corpus from the single-document summarization task at the Document Understanding Conference 2002^{xi}. This benchmark dataset contains 533 news articles, each accompanied by two to three human-generated *abstracts* of approximately 100 words each. Though DUC 2002 still serves as a widely recognised benchmark for single-document summarization, the best ROUGE-1 score (Lin & Hovy, 2003) achieved on it by a supervised system is only 50.11% (Mihalcea, 2005). Since supervised summarization systems learn from the human summaries, this result indicates the low degree of actual inter-assessor agreement, implying that the human assessors participating in the experiment had a poor understanding of their summarization task and interpreted it in a variety of ways. Another important implication of the above result, confirmed by the published performance of state-of-the-art summarization methods (e.g. TextRank, see Mihalcea, 2005, and MUSE, see Litvak & Last, 2012), is that no automated summarization system can be expected to exceed the 50 percent recall on this corpus.

For the Arabic language, a corpus compiled from 90 news articles was generated by Litvak and Last (2012). Each article was summarized by three native Arabic speakers, who selected the most important

sentences into an *extractive* summary of approximately 100 words each. All assessors were provided with the *Tool Assisting Human Assessors (TAHA)* software tool, which allowed the assessors to easily select the sentences to be included in the document extract. On this relatively small corpus, the agreement between assessors measured by the ROUGE-1 score reached 75 percent on average.

For the Hebrew language, Litvak and Last (2012) built a corpus of 120 news articles having 250 to 830 words each from the websites of the *Haaretz* newspaper^{xii}, *The Marker* newspaper^{xiii}, and manually translated articles from *WikiNews*^{xiv}. The articles were summarized by human assessors using the TAHA software. In total, 126 undergraduate students participated in the experiment. Each participant was randomly assigned ten different documents and instructed to choose the most important sentences in each document subject to the following constraints: (a) spend at least five minutes on each document, (b) ignore dialogs and quotations, (c) read the whole document before beginning sentence extraction, (d) ignore redundant, repetitive, and overly detailed information, and (e) remain within the minimal and maximal summary length limits (95 and 100 words, respectively).

Summaries were assessed for quality by comparing each student’s extract to those of all the other students using the ROUGE evaluation toolkit and the ROUGE-1 metric. The summaries of the assessors who received an average ROUGE score below 0.5, i.e. who agreed with the rest of the assessors in fewer than 50% of cases, were removed from the corpus. Also, the time spent by an assessor on each document was checked (with respect to the requirements). The final corpus of summarized Hebrew texts was compiled from the summaries of about 60% of the assessors, with an average of five *extracts* per single document. The average ROUGE score of the selected assessors was 54 percent only, only slightly higher than the quality of the DUC 2002 corpus.

The above-mentioned results in three different languages pose a severe limit on the maximum achievable accuracy of automated summarization systems. Consequently, user satisfaction with the output of these systems is expected to be low as well. As long as these subjective evaluation methods remain in use, a significant breakthrough in the single-document and multi-document summarization technology is highly unlikely. Graph-based interfaces may be a key to enhancing the consistency of “gold standard” summaries and improving the usability of text summarization systems.

3.3 Summary representation formats and cognitive (re)representation formats

As described in Section 2, there is an increasing multitude of formats in summarization. This raises the question which ones are “better” than others, i.e. how to cross-evaluate them. While we have described the increasing move away from sentential summaries (natural-language texts), particularly in the news summarization domain, in this section we want to argue why for cross-evaluation, we believe we need to go back to sentences. In particular, we propose a hypothesis about the cognitive representation of such summaries. The latter served as a starting point for our evaluation framework, presented next.

Advocating, on the one hand, non-sentential, graph-based summaries and arguing, on the other hand, for text as a common format for cross-evaluation, may seem like a paradox. We will come back to this and propose steps towards a resolution in Section 4.

News texts are typically (factual) *narrative texts*: temporal sequences of events, described in terms of answers to the “W” questions: Who? When? Where? What? The focus of the text is typically on a series of actions. So a good summary of one news text or several news texts is likely to focus on the most relevant actions/events, i.e. itself be a narrative. The “W”s at first sight appear to suggest a tabular format, but a closer look at the multitude of different real-life events (i.e. in particular the “what”) suggests that the richness of natural language is more appropriate. We therefore hypothesise that human news consumers, the ultimate target group of summaries, are likely to “translate” news summaries of whatever form into sentences in their head anyway, and that this translation will determine their actual mental model. In order to compare automatic summarization techniques based on this assumption, we therefore need to transform non-textual summary elements or patterns into sentences. The most direct way of doing this is to *use the patterns as selectors of sentences* – thus we *extend* the patterns into a form of extractive summarization.^{xv} To assess the quality of this (re-)construction of presumed novel “facts” in the story, we

then build on existing techniques for cross-evaluating sentential summaries to compare specific TTM methods with one another.

While we believe that ultimately, a news-summary user’s “translation” into sentences will determine this user’s mental model of the story, it is quite possible that different formats offered by automatic summarizers support this construction in different ways. *The automated cross-evaluation cannot demonstrate the superiority of a pattern type.* Therefore, we complement the automated cross-evaluation framework by a user-centric cross-evaluation of methods. Thus, we now let human end-users perform the reconstruction-by-selection that we modelled in the first stage. Here, the algorithms can no longer be evaluated in isolation but as part of an interactive user interface, and the evaluation criteria need to be extended by usability criteria.

4 HOW CAN DIFFERENT FORMATS BE COMPARED? A FRAMEWORK FOR CROSS-EVALUATING TTM METHODS

In this section, we will describe the procedures for automatic and for user-assisted evaluations of news summaries in a graph-based format. The general schema for automated evaluation, together with results from a case study, have first been presented in (Subašić & Berendt, 2010c, 2013). The general schema for user-centric evaluations is presented in Section 4.2. Results from an instance of the general schema for user-centric evaluation are presented in Section 5.

4.1 Cross-evaluating TTM methods: Automated approaches

The evaluation procedure is composed of the following steps: (a) generation of interesting patterns, (b) generation of a “ground truth”, (c) pattern-to-query transformation, (d) sentence retrieval using generated queries, and (e) comparison of retrieved sentences and editor-selected “fact” sentences. For step (a), we apply the respective TTM methods, which regard bursty patterns as interesting, see Section 3, or frequency as a baseline form of interestingness in patterns, see Section 5.1. Steps (b)-(e) are explained in the present section.

A note on temporal indexing that concerns all of the following subsections: we use techniques for generating a summary of events in a given period t within a story. Thus, the patterns are generated to be bursty within t , the documents are grouped into sets belonging to t , and the resulting retrieved sentences and editor-selected sentences likewise belong to t . In principle, any temporal structure could be applied; for simplicity, in our evaluation study we partitioned the whole story time into periods of equal length.

4.1.1 Crowdsourcing the ground truth?!

To compile ground-truth descriptions for corpora, we turned to the Web itself. The Web not only provides us with a wealth of documents, but also with summaries of events and their progression. Such “timeline” documents are particularly interesting when they have already undergone a process of testing for and creating a form of consensus – i.e. a form of inter-rater agreement. Documents authored in Wiki fashion and in particular Wikipedia articles, which are often written and re-written by hundreds of people, are good examples. While there is often extensive disagreement in Wikipedia (Viégas, Wattenberg, & Dave, 2004), it is generally acknowledged that text portions that remain in a Wikipedia article represent consensus (Kriplean, Beschastnikh, McDonald, & Golder, 2007; Adler, de Alfaro, Pye, & Raman, 2008) – with the big advantage that this is not a consensus between two or three raters, but between a much larger number of people. We therefore chose, wherever possible, this form of “crowdsourcing the ground truth” to create our set of ground-truth sentences. To maintain the time-indexed nature of temporal text mining, only sentences that contained a date specification (identified via a set of regular expressions) were selected from these documents. This approach works very well for many stories of common interest, see (Subašić & Berendt, 2010a, 2013).

Of course, this approach is not without problems. First, a story has to have raised enough interest to have inspired authors to write and publish a timeline document. Second, these documents are subject to the same quality challenges as other Wikipedia entries. Third, their authors have not written “their” parts independently, such that it is not straightforward to measure inter-rater agreement with the usual metrics. Fourth, it is difficult to account for authors’ self-selection bias (although problems of self-selection can also not be ruled out when using the standard methods of assessor recruitment). Finally, the information sources of the Wikipedia authors may be different from the article corpus used by a TTM method. We observed two cases in which the Wikipedia-based approach fails or in which other timeline documents are superior. The first case is a story in which developments happen at a certain time, but get discovered (and reported) only much later. The resulting reporting, in the timeline, of the dates on which events happened cannot serve as a summary of the reporting documents that appeared at this time. An example of this case was the Enron story (Wikipedia contributors, 2013). We decided not to cover such stories. The second case are special-interest stories for which more detailed sources exist. We found this to be the case for celebrity reporting and chose the timeline provided by a fan site as the ground truth. In sum, we believe that this method of defining a ground truth cannot replace others, but that it can be a useful complement. Future work should investigate to what extent this method can help avoid the problems mentioned above in Section 3.2.

4.1.2 From patterns to sentences: query generation and sentence retrieval

Due to differences in the expressiveness levels of story representations, comparing the patterns directly with editor-selected sentences would be biased towards the patterns with sentence-like structure. Therefore, to make direct comparisons possible, we developed a process for identifying the sentences that story elements resemble best. This task is akin to that of sentence retrieval: given a query, rank sentences based on some measure of their similarity to that query. Our approach is therefore to transform the patterns into queries and then use sentence retrieval. Direct comparisons are then possible on the retrieved sentences. The process is shown, along with an example for the combo method STORIES, in Fig. 1. For keyword-based methods, word conjunctions are formed; for group-based methods, conjunctions are formed from words within a group. We used the Query-likelihood retrieval method (QL) with Jelinek-Mercer topic smoothing for sentence retrieval, as suggested by the use of this method as a baseline by Murdock and Croft (2005).

Figure 1. The evaluation pipeline (top) and an example (bottom). The example turns a STORIES pattern (the subgraph highlighted in red constitutes the query) into a set of retrieved sentences.

The inputs for this model are an index of pseudo-documents and a set of queries used for retrieval. We use the set of all sentences from the complete document set as the index. Queries are obtained from story representations of evaluated TTM tracking methods.

We form two types of queries. *Generic query generation* uses the top-ranking bursty elements from each story-representation method. *Specific query generation* combines the basic story elements into more complex queries to account for the semantics of different story representations. An example are subgraphs-as-queries for the STORIES method, as shown in the example in Fig. 1.

For a given TTM method, we use each generic query to retrieve the top-ranked sentences using QL, as well as each specific query. This creates two sets of retrieved sentences per method.

4.1.3 Quality measures: Comparing retrieved “facts” with editor-selected “ground-truth” sentences

The quality of such a summary can be assessed by the degree to which the retrieved sentences (the presumed novel “facts”) resemble the “ground-truth” sentences. The challenge is to measure an aggregate reconstruction quality over the possible/plausible fact constructions. We compare retrieved sentences with ground-truth sentences in an IR-style evaluation employing measures for scoring each TTM method.

The “recall-oriented” aggregate measure $maxMR$ measures how well a ground-truth sentence is recalled. It uses an atomic measure sim of sentence-to-sentence fit (ROUGE2 or ROUGE.SU4, although others are also possible), and then uses the whole corpus to normalise this, accounting for differences in the number of retrieval candidates over periods. For each ground-truth sentence, the best-fitting sentence is found in the sentences retrieved by the query. The best *possible* fit is the maximum over all sentences indexed with the same time period. Formally:

$$maxMR_t = avg_{s \in G_t} (max_{r \in R_t} sim(r, s) / max_{r \in A_t} sim(r, s)),$$

where G are the ground-truth sentences, A all sentences from the documents, and $R \subseteq A$ the sentences retrieved by the bursty patterns of the method under consideration, each for period t .

The “precision-oriented” aggregate measure $maxMP$ scales $maxMR$ to reward methods that retrieve fewer sentences:

$$maxMP_t = maxMR_t \times min(|G_t|, |R_t|) / |R_t|.$$

Each method’s overall scores are formed by aggregating their $maxMR_t$ and $maxMP_t$ over all periods t . (The aggregation method can for example be an average if time is partitioned into equal-sized periods.) With the help of the Friedman/Tukey multiple comparison test (Kendall, 1976), we can then determine, for a group of methods, which ones differ from one another in a statistically significant way and which ones do not, thus identifying a *group* of top performers. We evaluated the methods named above and varied a number of parameters including how many of the top-ranking patterns were used for sentence retrieval. For example, methods that belong to the top performers even when only their top 5 patterns are used, can be considered to be high-precision in more senses than one: a small number of their top patterns retrieves a small number of sentences, and these do well at recalling relevant ground-truth sentences.

4.1.4 Case study: An automated evaluation of several TTM methods

In (Subašić & Berendt, 2010c, 2013), we presented a case study of the above evaluation framework that evaluated three methods over five corpora: (Kleinberg, 2002) as a representative of keyword representations, (Mei & Zhai, 2005) as a representative of group representations, and STORIES (Subašić & Berendt, 2010a) as a representative of combo representations (see Section 2.2 for a definition of the three types of representation). The study varied a number of parameters for pattern generation and sentence retrieval. The results suggest that the STORIES method is the most robust of all tested methods. The research also shows that using bursty patterns’ internal structure for connecting them with ground truth sentences improves the results.

4.1.5 Limitations

Any given TTM method constructs a certain pattern type *and* extracts certain information from a corpus. Through the proposed mapping to a *common* format, namely sentences, we were able to make different TTM methods comparable. This automated comparison thus focused on the extracted information and – by construction – abstracted from the differences in pattern type.

The automated comparison cannot demonstrate the superiority of a pattern type also for a second reason: For any given extracted information (here operationalised as a given set of retrieved sentences R^*) and

any given pattern type, one can define a TTM algorithm that generates patterns of this type that will retrieve R^* .^{xvi}

Thus, such comparisons cannot answer an important question that a system designer would ask: Are certain pattern types better for humans? Do they present better affordances for the (re)construction of events, do they give people faster insights into a story, do they lead to a more enjoyable user experience? The usefulness of the automated comparison method described above also strongly depends on the quality of the “ground-truth” sentences selected by the editors or the annotators. As shown in Section 3.2 above, this can be the weakest link in the evaluation pipeline.

In the next section, we will therefore propose a complementary cross-evaluation methodology. We will argue that in user-centric evaluations, we can fix the algorithm and thus the information presented and vary the pattern type.

4.2 Cross-evaluating TTM methods and interfaces: User-centric approaches

Various approaches to (news-)story tracking led to the development of different research areas like update summarization and temporal text mining. In this section and the following one, we focus on the users and investigate document-search interfaces for story tracking. We propose a set of tasks and measures to evaluate document-search interfaces for story tracking. To the best of our knowledge, this is the first cross-evaluation study of document-search interfaces in the context of story tracking, and the first systematic account of its relation to automated cross-evaluation.

Many tasks related to information retrieval have been approached with interaction in mind (Ruthven, 2008), and several of these tasks have been assessed using standardised evaluation frameworks (Dumais & Belkin, 2005). However, all of these disregard the temporal dimension of corpora and thus are not fully appropriate for interactive story tracking evaluation. We transform the story tracking task into a fact finding and summary creation task. On one side this is similar to the fact finding tasks of the HARD track (Allan, 2004), but we provide a broader description of the topics and include search over several time periods.

In the following, we describe the specific choices made in our evaluation of different aspects of our TTM method STORIES (see Section 2.3), but frame this specific user experiment in terms of a general scheme for cross-evaluating TTM methods.

4.2.1 From patterns to sentences: the user task

We framed the story tracking task as an interactive IR task of fact finding and summary creation. Patterns were given to users as summaries and document-search interfaces. We asked them to identify sentences from the documents that “best describe the time period they belong to”. We refer to the resulting interactive interfaces as *summarization interfaces*. To simulate story tracking, the task should be solved in a number of consecutive time periods.

4.2.2 Quality measures: summary quality and other indicators of usability and usefulness

We defined five sets of measures to capture the various aspects of interfaces for story tracking. The first two are observation-based, the remaining ones are based on self-reports and assess subjective evaluations of various aspects of usability and usefulness of the summarization interfaces in relation to the task.

Summary quality measures. We quantify the summary quality using the measures for automatic story tracking evaluation described above. For each user and period, $maxMR$ is calculated.^{xvii}

User activity measures. Activity measures show the level of user engagement in story tracking. We wish to investigate whether different summarization interfaces provide an incentive for corpus exploration. We

defined four activity measures: query length (number of words), number of issued queries, number of accessed documents, and exploration time.

Other usability and usefulness measures. Questionnaires were used to assess users’ subjective impressions of usability and usefulness. Questions centring on the summarization interface asked how well the summarization interface supports the following activities: quick scan of the documents, exploration of the different aspects of the document set, and discovery of relevant documents. Other questions refer to the task and users’ own performance within the summarization interface: how tedious and overwhelming the task was, and how they perceived their success in solving the task. We also requested a ranking of the summarization interfaces based on four criteria: easy to learn, easy to use, suitable to solving the task, and personal preference.

5 USER-CENTRIC EVALUATION OF STORY TRACKING METHODS: A CASE STUDY

In this section, we describe a detailed example of the user-centric cross-evaluation described in general terms in the previous section.

5.1 Pattern types, tested aspects, and their operationalisation as summarization interfaces

To compare different TTM methods, one needs to offer them in interfaces that resemble one another as much as possible. We presented users with four different summarization interfaces (SIs). One was the STORIES interface, in which users interact with the story graph by selecting edges that they find relevant. These edges are then transformed into a query by concatenating the names of nodes they connect. The system returns a set of relevant documents for this query. The main components of STORIES are temporal patterns that we use as *suggestions* and visual graph-based *search*. (Recall from Fig. 1 that a search query is submitted by marking a subgraph.) To test these components, we created two further SIs: GRAPH, a graph-based visual search interface based on non-temporal patterns, and SUGGEST, a text-based search using temporal patterns presented as a list. We added a standard keyword-based interface (S.BOX) as a baseline SI. Figure 2 shows the SIs we tested. GRAPH and STORIES look the same, and S.BOX and SUGGEST differ only in the presence of a bi-gram list (marked by a red rectangle in the figure).

Figure 2. Screen shots of the interfaces: (a) graph-based interfaces STORIES and GRAPH; (b) text-based interfaces SUGGEST and S.BOX. SUGGEST and S.BOX are identical except for the highlighted area (marked with a red rectangle), which is visible only in SUGGEST.

Generating suggestions and graphs. We generated two types of suggestions, temporal and non-temporal. First, we extracted the content-bearing terms, defined as the 150 top-TF.IDF terms. The suggestions were generated for a time period t as described in Section 2.3. For the temporal suggestions used in STORIES and SUGGEST, the TR measure was used for the edge weights, and for the non-temporal in GRAPH, the LR measure was used. Finally, we sorted the edges by weight and kept the top 30.

5.2 Method

Participants. We recruited 24 participants (9 female) using a student forum. As an incentive for participation, we offered a 20 Euros voucher for a retail store. Participants had a wide range of study directions, and most (16) were PhD or Master students.

Materials. The corpora we collected each consist of two parts: the document set and the ground-truth set. The document set is divided into stories which are the sub-corpora discussing the same news subjects. The subject is regarded as a higher-level news story. In total, we compiled four stories covering: Britney Spears, the Greek debt crisis, the BP oil spill, and the Chilean miners' accident. A ground-truth set was created for each story, using the technique described in Section 4.1.1. Participants explored the document sets using the four SIs described in Section 5.1.

Design. Each user tested all SIs and was assigned a unique ordering of them. We counterbalanced the order by generating all possible orderings of the four interfaces. There was no reason to expect an effect of the order of the stories; therefore, this was kept constant. This produced a total of 24 orderings. Using a single SI, a participant always explored the same news story.

Procedure. The participants were asked to read the task instruction and solve the task of identifying sentences from the documents that “best describe the time period they belong to”. To simulate story tracking, the task was solved in three consecutive time periods. Before using each SI, participants were provided with a tutorial followed by a test run. For each period, participants were presented with the interface and the initial document list sorted by date and relevance. A time limit of seven minutes per period was set, following a pilot study in which participants completed the task without the time limit. After seven minutes, participants could finish reading the document they were inspecting at this moment and select sentences from it, but neither search for nor read further documents.

Figure 3: Study results for the observed measures, summary quality (top) and user activity (bottom left); and comparative measures (bottom right).

Before solving the tasks, participants filled in a demographics pre-study questionnaire, and after each SI they filled in a post-system questionnaire, in which they rated the interface-specific “other usability and usefulness measures” described in Section 4.2.2 on a five-point Likert scale. At the end of the study (after the fourth SI), participants were given an exit questionnaire in which they were asked to compare all the SIs they had used. Together with this exit questionnaire, participants were presented with the screenshots of the SIs in the same order as they had been used during the study.

Participants were tested in individual sessions lasting about two hours each. After the second SI test was completed, there was a 15 minute break. The participants worked with a Mac minicomputer using a 17” monitor, with the Web-based software and documents residing on a server.

5.3 Results

The value for all summary quality measures (Fig. 3a) was always highest for the graph-based interfaces (GRAPH and STORIES). We tested the differences using the Kruskal-Wallis test (due to the non-normal distribution used instead of ANOVA) with Tukey’s HSD correction for multiple comparisons. No statistically significant differences in summary quality were found between interfaces. The analysis of user activity measures (Fig. 3b) showed that on average, participants used more and longer queries with GRAPH and STORIES. The longest queries (avg. 2.33) were issued with STORIES and the shortest (1.7) with S.BOX. The differences were statistically significant ($p < .05$) between all interfaces except between GRAPH and STORIES. With the average of 10.4 queries per period, STORIES engaged participants to

issue the highest number of queries. The fewest queries were issued using S.BOX (4.84). The differences between S.BOX and all other interfaces were statistically significant. As expected, issuing more queries resulted in more document accesses. With STORIES, participants on average read around 27 documents. The only statistically significant difference was between STORIES and S.BOX. We found no differences between interfaces in exploration time.

Table 1. Average values of self-reported measures.

The self-reported measures were analysed assuming equal distances between the scores on the Likert scale; averages are shown in Table 1. On average, users found that the graph-based interfaces provided for an *easier access* to the relevant documents and *exploration* of aspects in a story. Participants rated the task as slightly more *tedious* to complete with graph-based interfaces. However, the differences between interfaces were not significant for any measure.

The results of interface comparison (Fig. 3c) show that participants ranked S.BOX as the *easiest for using and learning*. On the other two criteria, users ranked GRAPH and STORIES as the *most fitting to the task* of the study, and they expressed *personal preference* towards them. All four criteria were tested using the Friedman test with Tukey’s HSD. With regard to the criterion *easy to learn*, only the differences between S.BOX and all others were significant ($p < .05$). For the criterion *easy to use*, the only significant difference of ranking was that between S.BOX and GRAPH. For both the criteria *fitting to the task* and *personal preference*, there was a significant difference between the ranks of GRAPH and STORIES and the ranks of SUGGEST and S.BOX.

5.4 Discussion

For all interfaces, we found no statistically significant differences in summary quality measures. The observed low performance of participants is most likely the cause of the lack of differentiation between the interfaces. We speculate that the reason for a low performance is the complexity of the task. Creating a summary may be a job more suited to professional journalists than to regular web users. Nevertheless, we discovered several valuable insights into interfaces for story tracking. First, we observed that with time, summary quality improves. In the last periods, summary quality was higher than in the first two by around 20% for GRAPH and STORIES. In addition, we found that GRAPH and STORIES engaged participants in deeper exploration of the document set: they issued more queries and longer queries, and they accessed more documents compared to other interfaces. All interfaces except S.BOX include query suggestions, and queries can be issued without typing. This provides an easier way of querying, and the study shows that users will use facilitated querying (clicking rather than typing) if provided.

For the self-reported measures, we found no statistically significant differences, but the analysis of the direct comparison of the interfaces shows that users expressed higher preference for using graph-based interfaces and ranked these two interfaces as the most fitting to the task of the study. The biggest differences were found between SUGGEST and STORIES/GRAPH. This suggests that out of graph representation and suggestion generation components, the first one is a more important one. This may indicate that temporal patterns are not more useful than non-temporal ones for graphs, or alternatively that the difference is due to the low performance on the tasks. As this study does not follow the full factorial design, this finding should be interpreted keeping this in mind. For most criteria, the standard search-box interface is outperformed by graph interfaces.

In sum, although text-based interfaces are still the most widely used interfaces for search today, we were able to show that graph-based interfaces can provide for a different, and for some users better, search experience without diminishing the task performance.

We recognise a number of limitations of this study. It is not clear whether the task we defined is a “good” simulation of story tracking. We asked participants to rate, on a five-point scale, how similar the defined

task is to the way they follow news, and the average rating was just over three (neutral). Another issue concerns current habits: although users had not seen the interfaces prior to the study, text-based interfaces resemble most commonly used search interfaces. Thus, some bias towards these interfaces was expected, as shown by user perception on learning and operating difficulty. Also, the study should be replicated with more users.

6 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this chapter, we have investigated the summarization of multiple news documents and the challenges this poses for evaluation. In particular, we have taken a close look at the role of the sentential format. We have described several new alternatives to presenting summaries as natural-language texts, looking in particular at graph-based formats. In this concluding section, we will summarize the main points and extend the discussion in order to outline directions for future work.

First, we have argued that to compare different approaches to summarization with one another, we need to map them to a common format. We have argued that the mental models that users construct of news texts are likely to be sentential and that therefore *a mapping of summaries to a sentential format is advantageous for the purposes of comparison*. However, such comparisons themselves rely on the existence and availability of a ground truth, which is also sentential – although studies have shown this to be a non-robust procedure of representing a common understanding of multiple texts. This shows that *a sentential format is also problematic for the purposes of comparison*.

Can the non-sentential formats that we have discussed offer a way out of this dilemma? The results of our user study suggest that it is essential for effective summarization to integrate the user into sense-making through usable, entertaining and ultimately useful interactive summarization-plus-document-search interfaces. Today’s news aggregators are just the beginning in this. In particular, we believe that interactive graphical summaries, such as STORIES, illustrate the potentials of future multi-document summarization. Other graph-based text representations, such as those used by MUSE, can also be adapted for the interactive summarization purposes.

This may be leveraged in two ways that focus on users in their roles as assessors and as end users, respectively. First, graph-based methods and interfaces may be a better *preparation* for people to concentrate on what is essential in a collection of texts, and thus may be a key to enhancing the consistency of “gold standard” summaries for evaluating various summarization tasks. Such high-quality summarization corpora should also enhance the performance of supervised summarization methods that heavily depend on the quality of their training data. Second, it is also conceivable that the quest for “the best” summarization is bound to fail because *there is no summary that is best for everyone*, that instead, the best summary is one which stimulates the end user to engage in interactive and individual sense-making. This perspective augments machine intelligence (through automatic summarization, query-dependence and user-adaptivity) by a focus on usability. Such a paradigm shift towards user-customised summaries should also change the way we evaluate text summarization techniques and replace the “one gold standard fits all” approach with carefully designed user studies built upon a variety of summary representation formats.

While graph-based interfaces can provide such usability, an outlook on future summarization should also consider the broader context of new devices and use cases. The increasing importance of mobile devices especially for the consumption of news presents an interesting test case. On the one hand, the limitations of screen resolution both in display and in the granularity of interactivity appear to favour text as a more “robust” representation format. A recent example is the success of the summarization app Summly, which was recently bought by Yahoo! (Winch, 2013). Summly uses a proprietary single-document extractive summarization algorithm, which compresses the content of news articles into 2-3 key sentences that fit onto an iPhone screen. The users can link to the original article if they like the summary. Another example is the summarization app based on MUSE^{xviii}.

The current popularity of short texts probably also results from habits of information consumption and production shaped by Twitter. On the other hand, devices are developing fast and becoming increasingly available in a variety of screen sizes. Also, visual information presentation is hugely popular on mobile devices, and it may often be easier to apprehend at a glance than textual information (“a picture says more than a thousand words”), especially when information is consumed on the move. We believe that this presents a good starting point for graph-based summarization formats to exploit and shape visual literacy for complementary and/or improved ways of presenting information on mobile devices.

These developments may also be accompanied by advances in summarization of other types of content, for which not even people’s mental models of summaries may be sentential. As an example, take sports reports. These are also narratives (and news), but for many purposes the information need may be much more schematic: who played against whom in what, and which score was the result? For such purposes, a tabular summary of key variables’ values may be optimal (such that the summary can easily be used to compare teams’ performance, create statistics, etc.), see Narrativescience^{xix} and (Levy, 2012). Appropriate formats may change even more when content in other modes of discourse is summarized. For example, for product reviews as a form of *argumentative/evaluative text*, the desired summarization may be an aggregate such as the relative number of positive vs. negative opinions on a product feature, calling for extractive-abstractive hybrids (Carenini & Cheung, 2008). For search results (which may be considered a form of *descriptive text*), the current user interfaces of Web search engines suggest that a list format, or a combination of list and aspect-oriented results, is desired. *Expository texts* such as scientific essays may require yet other contents and formats of summarization. The extension of the analysis presented in this paper to different types of summarization is an exciting area of future research. Such further analysis should assist in choosing the best summarization interfaces and formats for different content types and user devices.

REFERENCES

- Adler, B. T., de Alfaro, L., Pye, I., & Raman, V. (2008). Measuring author contributions to the Wikipedia. In *Proceedings of WikiSym '08*. (pp. 15:1-15:10). New York: ACM.
- Allan, J. (Ed.). (2002). *Topic Detection and Tracking: Event-Based Information Organization*. Norwell, MA, USA: Kluwer Academic Publishers.
- Allan, J. (2004). HARD track overview in TREC 2004 – high accuracy retrieval from documents. In E. M. Voorhees & L. P. Buckland (Eds.), *TREC (Vol. Special Publication 500-261)*. Gaithersburg, MD: National Institute of Standards and Technology (NIST).
- Allan, J., Gupta, R., & Khandelwal, V. (2001). Temporal summaries of news topics. In W. B. Croft, D. J. Harper, D. H. Kraft, & J. Zobel (Eds.), *Proc. SIGIR* (pp. 10-18). New York: ACM.
- Allan, J., Lavrenko, V., & Jin, H. (2000). First story detection in TDT is hard. In *Proc. 9th CIKM* (pp. 374-381). New York: ACM.
- Barzilay, R., McKeown, K., & Elhadad, M. (1999). Information fusion in the context of multi-document summarization. In R. Dale & K. W. Church (Eds.), *Proc. ACL*. (pp. 550-557). Stroudsburg, PA, USA: Association for Computational Linguistics.
- Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30, 107-117.

- Carenini, G., & Cheung, J. C. K. (2008). Extractive vs. NLG-based abstractive summarization of evaluative text: The effect of corpus controversiality. In M. White, C. Nakatsu, & D. McDonald (Eds.), *Proc. INLG*. Stroudsburg, PA, USA: Association for Computational Linguistics.
- Chen, C. C., & Chen, M. C. (2008). TSCAN: a novel method for topic summarization and content anatomy. In *Proc. 31st SIGIR* (pp. 579-586). New York: ACM.
- Dumais, S. T., & Belkin, N. J. (2005). The TREC interactive tracks: Putting the user into search. In *Proc. TREC'05* (pp. 22-31). Gaithersburg, MD: National Institute of Standards and Technology (NIST).
- Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22, 457-479.
- Fung, G. P. C., Yu, J. X., Yu, P. S., & Lu, H. (2005). Parameter free bursty events detection in text streams. In *Proc. VLDB* (pp. 181-192). New York: ACM.
- Gillick, D., & Favre, B. (2009). A scalable global model for summarization. In *Proceedings of the Workshop on integer linear programming for natural language processing* (pp. 10-18). Stroudsburg, PA, USA: Association for Computational Linguistics.
- Gruhl, D., Guha, R. V., Kumar, R., Novak, J., & Tomkins, A. (2005). The predictive power of online chatter. In *Proc. SIGKDD* (pp. 78-87). New York: ACM.
- Hahn, U., & Mani, I. (2000). The challenges of automatic summarization. *IEEE Computer*, 33 (11), 29-36.
- He, Q., Chang, K., Lim, E.-P., & Zhang, J. (2007). Bursty feature representation for clustering text streams. In *Proc. SDM*. Minneapolis, MN: SIAM. Retrieved July 13, 2013 from http://www.siam.org/proceedings/datamining/2007/dm07_050he.pdf
- Janssens, F. A. L., Glänzel, W., & De Moor, B. (2007). Dynamic hybrid clustering of bioinformatics by incorporating text mining and citation analysis. In *Proc. SIGKDD* (pp. 360-369). New York: ACM.
- Kendall, M. (1976). *Rank correlation methods*. Oxford: Oxford University Press.
- Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46 (5), 604-632.
- Kleinberg, J. (2002). Bursty and hierarchical structure in streams. In *Proc. SIGKDD* (pp. 91-101). New York: ACM.
- Kriplean, T., Beschastnikh, I., McDonald, D. W., & Golder, S. A. (2007). Community, consensus, coercion, control: CS*W or how policy mediates mass participation. In *Proceedings of the 2007 ACM Conference on Supporting Group Work (GROUP '07)* (pp. 167-176). New York: ACM.
- Levy, S. (2012). Can an algorithm write a better news story than a human reporter? Wired. Retrieved July, 13, 2013 from <http://www.wired.com/gadgetlab/2012/04/can-an-algorithm-write-a-better-news-story-than-a-human-reporter/>
- Lin, C.-Y., & Hovy, E. (2003). Automatic evaluation of summaries using N-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for*

Computational Linguistics on Human Language Technology (NAACL '03) (pp. 71-78). Morristown, NJ, USA: Association for Computational Linguistics.

Litvak, M., Aizenman, H., Gobits, I., Last, M., & Kandel, A. (2011). DegExt: A language-independent graph-based keyphrase extractor. In *Proceedings of the 7th Atlantic Web Intelligence Conference (AWIC 2011)* (pp. 121-130). Berlin: Springer.

Litvak, M., & Last, M. (2008). Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multi-Source Multilingual Information Extraction and Summarization* (pp. 17-24). Stroudsburg, PA, USA: Association for Computational Linguistics.

Litvak, M., & Last, M. (2012). Cross-lingual training of summarization systems using annotated corpora in a foreign language. *Information Retrieval*. DOI <http://dx.doi.org/10.1007/s10791-012-9210-3>

Long, C., Huang, M., Zhu, X., & Li, M. (2009). Multi-document summarization by information distance. In *Proc. ICDM 2009* (pp. 866-871). Silver Spring, MD: IEEE Computer Science Press.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2, 159-165.

Luo, G., Tang, C., & Yu, P. S. (2007). Resource-adaptive real-time new event detection. In *Proc. SIGMOD 2007* (pp. 497-508). New York: ACM.

Mei, Q., & Zhai, C. (2005). Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proc. SIGKDD* (pp. 198-207). New York: ACM.

Mihalcea, R. (2005). Language independent extractive summarization. In *Proc. AAAI* (pp. 1688-1689). AAAI Press.

Murdock, V., & Croft, W. B. (2005). A translation model for sentence retrieval. In *Proc. HLT/EMNLP*. Stroudsburg, PA, USA: Association for Computational Linguistics.

Nallapati, R., Feng, A., Peng, F., & Allan, J. (2004). Event threading within news topics. In *Proc. CIKM 2004* (pp. 446-453). New York: ACM.

Nenkova, A., Passonneau, R., & McKeown, K. (2007, May). The Pyramid Method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing*, 4 (2), Article No. 4.

Petrović, S., Osborne, M., & Lavrenko, V. (2010). Streaming first story detection with application to Twitter. In *HLT '10: Human language technologies: The 2010 Annual Conf. of the North American Chapter of the Association for Computational Linguistics* (pp. 181-189). Morristown, NJ, USA: Association for Computational Linguistics.

Ruthven, I. (2008). Interactive information retrieval. *Annual Review of Information Science and Technology*, 42 (1), 43-91.

Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997). Automatic text structuring and summarization. *Information Processing and Management*, 33 (2), 193-207.

- Schult, R., & Spiliopoulou, M. (2006). Discovering emerging topics in unlabelled text collections. In *Proc. ADBIS* (pp. 353-366). Berlin: Springer: LNCS 4152.
- Smith, D. A. (2002). Detecting and browsing events in unstructured text. In *Proc. SIGIR* (pp. 73-80). New York: ACM.
- Soboroff, I., & Harman, D. (2005). Novelty detection: the TREC experience. In *HLT '05: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 105-112). Morristown, NJ, USA: Association for Computational Linguistics.
- Steinberger, J., & Ježek, K. (2009). Update summarization based on novel topic distribution. In *Proceedings of the 9th ACM Symposium on Document Engineering* (pp. 205-213). New York: ACM.
- Subašić, I., & Berendt, B. (2010a). Discovery of interactive graphs for under-standing and searching time-indexed corpora. *Knowledge and Information Systems*, 23 (3), 293-319.
- Subašić, I., & Berendt, B. (2010b). Experience stories: A visual news search and summarization system. In *Proc. ECML/PKDD* (3) (pp. 619-623). Berlin: Springer. LNCS 6323.
- Subašić, I., & Berendt, B. (2010c). From bursty patterns to bursty facts: The effectiveness of temporal text mining for news. In *Proc. ECAI* (Vol. 215, pp. 517-522). Amsterdam, The Netherlands: IOS Press.
- Subašić, I., & Berendt, B. (2013). Story graphs: Tracking document set evolution using dynamic graphs. *Intelligent Data Analysis*, 17 (1), 125-147.
- Trampus, M., & Mladenic, D. (2009). Constructing event templates from written news. In *Proc. Web Intelligence/IAT Workshops* (pp. 507-510). Silver Spring, MD: IEEE Computer Science Press.
- Turney, P. D. (2000). Learning Algorithms for Keyphrase Extraction. *Information Retrieval*, 2 (4), 303-336.
- Viégas, F. B., Wattenberg, M., & Dave, K. (2004). Studying cooperation and conflict between authors with history flow visualizations. In *Proc. CHI* (pp. 575-582). New York: ACM.
- Wang, X., & McCallum, A. (2006). Topics over time: a non-Markov continuous-time model of topical trends. In *Proc. SIGKDD* (pp. 424-433). New York: ACM.
- Wikipedia contributors (2013, May 28). Enron scandal. In *Wikipedia, The Free Encyclopedia*. Retrieved 13, 2013, from http://en.wikipedia.org/w/index.php?title=Enron_scandal&oldid=545449463
- Winch, J. (2013). British teen sells Summly app for millions. *The Telegraph*. Retrieved March, 25, 2013 from <http://www.telegraph.co.uk/finance/newsbysector/mediatechnologyandtelecoms/9952658/British-teen-sells-Summly-app-for-millions.html>

ADDITIONAL READING SECTION

- Balasubramanian, N., Allan, J., & Croft, W. B. (2007). A comparison of sentence retrieval techniques. In *Proc. 30th SIGIR* (pp. 813-814). New York: ACM.

- Barzilay, R., Elhadad, N., & McKeown, K. (2002). Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17, 35-55.
- Bollegala, D., Okazaki, N., & Ishizuka, M. (2006). A bottom-up approach to sentence ordering for multi-document summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (ACL-44)*. Association for Computational Linguistics, (pp. 385-392). Stroudsburg, PA, USA.
- Bysani, P. (2010). Detecting novelty in the context of progressive summarization. In *Proceedings of the NAACL HLT 2010 Student Research Workshop (HLT-SRWS '10)*. Association for Computational Linguistics, (pp. 13-18). Stroudsburg, PA, USA.
- Filippova, K. (2010). Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)*. Association for Computational Linguistics, (pp. 322-330). Stroudsburg, PA, USA.
- Giannakopoulos, G., & Karkaletsis, V. (2010). Summarization System Evaluation Variations Based on N-Gram Graphs. *TAC 2010 Proceedings Papers*.
- Li, T., & Shen, C. (2010). Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)* (pp. 984-992). Association for Computational Linguistics, Stroudsburg, PA, USA.
- Lin, H., & Bilmes, J. (2010). Multi-document Summarization via Budgeted Maximization of Submodular Functions. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10)* (pp. 912-920). Stroudsburg, PA, USA: Association for Computational Linguistics.
- McKeown, K., Passonneau, R. J., Elson, D. K., Nenkova, A., & Hirschberg, J. (2005). Do summaries help? In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '05)*, (pp. 210-217). New York, NY, USA.
- Murdock, V., (2006). Aspects of sentence retrieval. Doctoral dissertation, University of Massachusetts Amherst, Amherst, Massachusetts. Retrieved July 13, 2013, from <http://maroo.cs.umass.edu/pdf/IR-542.pdf>
- Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In C. C. Aggarwal & C. Zhai (Eds.), *Mining Text Data* (pp. 43-76). New York: Springer.
- Saggion, H. (2006). Multilingual Multidocument Summarization Tools and Evaluation. *International Conference on Language Resources and Evaluation (LREC)*.
- Schlesinger, J. M. (November 14-15, 2011). CLASSY 2011 at TAC: Guided and Multi-lingual Summaries and Evaluation Metrics. *Proceedings of the Fourth Text Analysis Conference (TAC 2011)*. Gaithersburg, Maryland, USA: National Institute of Standards and Technology.
- Wan, X. (2010). Towards a unified approach to simultaneous single-document and multi-document summarizations. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)* (pp. 1137-1145). Stroudsburg, PA, USA: Association for Computational Linguistics.

- Wan, X., & Yang, J. (2008). Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08)* (pp. 299-306). ACM.
- Wan, X., Li, H., & Xiao, J. (2010). Cross-language document summarization based on machine translation quality prediction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)* (pp. 917-926). Stroudsburg, PA, USA.
- Wang, D., & Li, T. (2010). Many are better than one: improving multi-document summarization via weighted consensus. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '10)*, (pp. 809-810). New York, NY, USA.
- Wang, D., Zhu, S., Li, T., & Gong, Y. (2009). Multi-document summarization using sentence-based topic models. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers (ACLShort '09)* (pp. 297-300). Association for Computational Linguistics, Stroudsburg, PA, USA.
- Wei, F., Li, W., Lu, Q., & He, Y. (2010). A document-sensitive graph model for multi-document summarization. *Knowledge and Information Systems*, 22(2), 245-259.
- Xiaodan, X. (2010). Study on Sub Topic Clustering of Multi-documents Based on Semi-supervised Learning. 2nd International Workshop on Database Technology and Applications. Wuhan, China: IEEE Computer Society.
- Xiong, Y., Liu, H., & Li, L. (2010). Multi-Documents summarization based on improved features and clustering. *International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE)*, (pp. 1-5).
- Zhang, J., Cheng, X., Xu, H., Wang, X., & Zeng, Y. (2008). Summarizing Dynamic Information with Signature Terms Based Content Filtering. Retrieved 09 06, 2011, from *TAC 2008 Proceedings Papers*. Retrieved November 21, 2012 from <http://www.nist.gov/tac/publications/2008/papers.html>
- Zhao, L., & Wu, L. (2009). Using query expansion in graph-based approach for query-focused multi-document summarization. *Information Processing & Management*, 45(1), 35-41.
- Zhou, L., & Lin, C. (2006). ParaEval: using paraphrases to evaluate summaries automatically. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL '06)* (pp. 447-454). Stroudsburg, PA, USA: Association for Computational Linguistics.

KEY TERMS & DEFINITIONS

Burst: sudden surge of importance of some element, where the importance of an element can be measured with frequency, probability, etc.

Bursty pattern: a text pattern from a story going through a burst.

Extractive summarization: selection of a subset of the most relevant fragments from a source text into the summary. The fragments can be paragraphs, sentences, keyphrases or keywords.

Generic Summarization: processes only a document (or a set of documents) to be summarized.

Language-independent sentence features: statistical features that do not require any morphological or syntactic analysis of the summarized text.

Language-specific sentence features: features that require dictionaries and Natural Language Processing (NLP) tools.

MUSE (MULTilingual Sentence Extractor): an extractive generic single-document summarization approach, which uses a linear combination of language-independent features for ranking each sentence in a text document.

Query-based Summarization: processes a document (or a set of documents) to be summarized as well as a query expressing the user's interest.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation): an automated method for comparing a computer-generated summary to gold standard summaries (generated by human annotators) in terms of overlapping units such as n-grams, word sequences, and word pairs.

Story: a set of time-stamped documents related to the same subject (e.g. all news reports about a specific natural disaster such as earthquake in Japan 2002).

Temporal text mining: a family of temporal text mining methods that outputs bursty text patterns (bursty patterns).

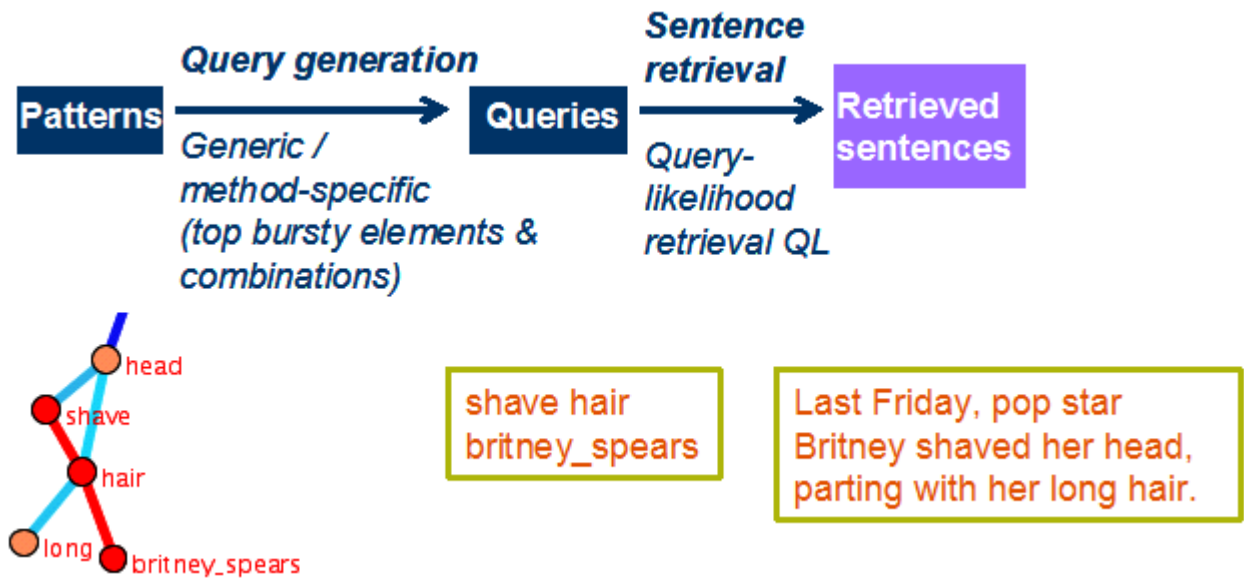


Figure 1. The evaluation pipeline (top) and an example (bottom). The example turns a *STORIES* pattern (the subgraph highlighted in red constitutes the query) into a set of retrieved sentences.



Figure 2. Screen shots of the interfaces: (a) graph-based interfaces STORIES and GRAPH; (b) text-based interfaces SUGGEST and S.BOX. SUGGEST and S.BOX are identical except for the highlighted area (marked with a red rectangle), which is visible only in SUGGEST.

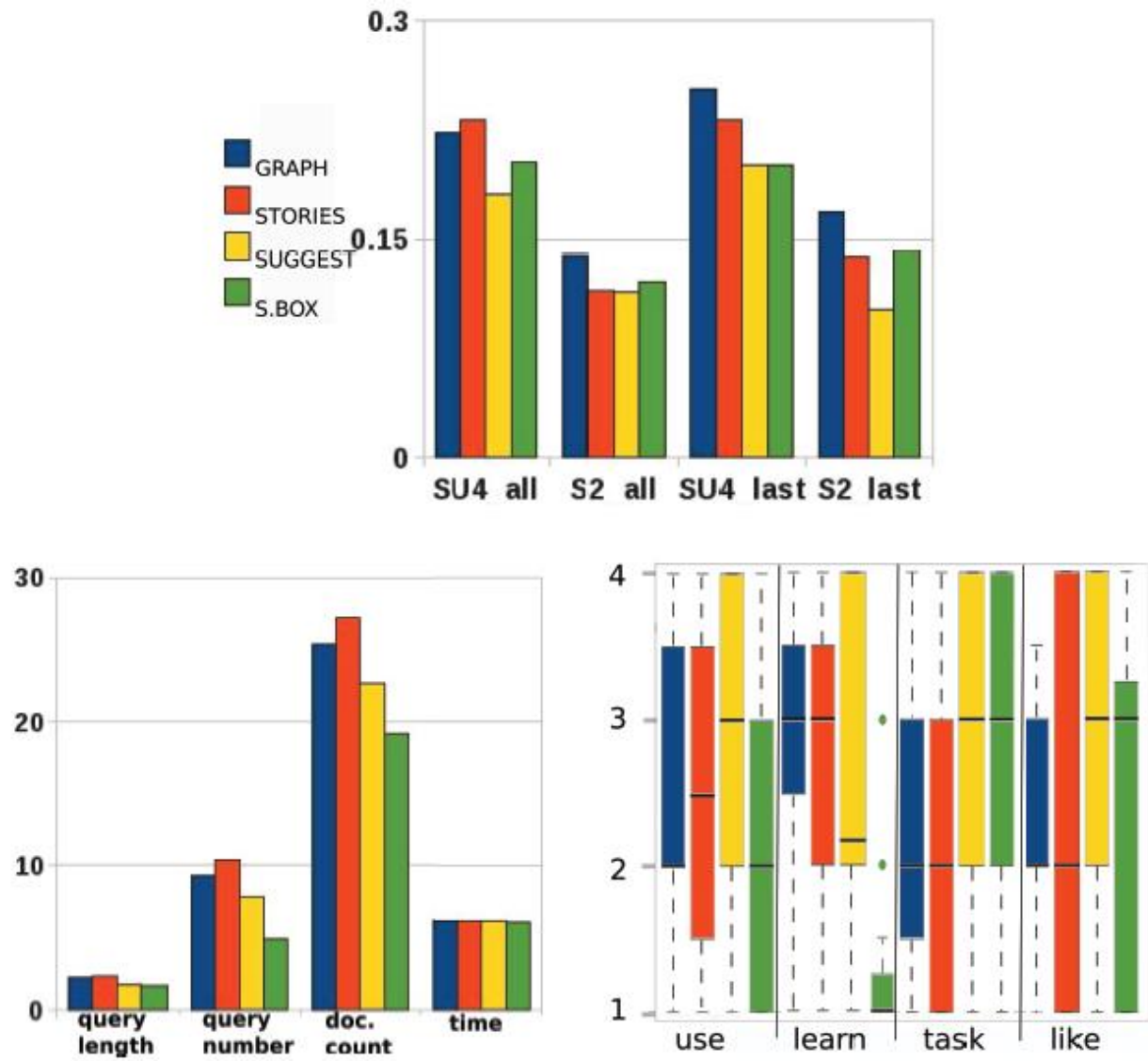


Figure 3: Study results for the observed measures, summary quality (top) and user activity (bottom left); and comparative measures (bottom right).

	GRAPH	STORIES	SUGGEST	S.BOX
quick scan	3.76	3.32	3.11	3.39
aspect exploration	3.62	3.32	3.06	3.13
relevant documents	3.81	3.55	3.22	3.35
overwhelming	2.95	2.91	2.56	2.17
tedious	2.67	2.86	2.83	2.65
successful	3.81	3.5	3.50	3.70

Table 1. Average values of self-reported measures.

ⁱ <http://news.google.com>

ⁱⁱ <http://news.yahoo.com>

ⁱⁱⁱ <http://multiling.iit.demokritos.gr/pages/view/662/multiling-2013/>

^{iv} In 2008, DUC became a Summarization track in the Text Analysis Conference (TAC).

^v <http://www-nlpir.nist.gov/projects/duc/duc2007/tasks.html#pilot>

^{vi} <http://www.trec-ts.org/home>

^{vii} Thus, in TDT evaluation, events correspond to story subjects. In that sense, FSD differs from story tracking in that it aims to detect documents discussing a new story subject. However, it is easy to imagine that all documents of an incoming stream belong to the same story. In this case the “new events” would correspond to developments making FSD more similar to our notion of the story tracking task.

^{viii} <http://www.opencalais.com/>

^{ix} <http://lucene.apache.org>

^x <http://duc.nist.gov/duc2004/protocol.html>

^{xi} <http://duc.nist.gov>

^{xii} <http://www.haaretz.co.il>

^{xiii} <http://www.themarker.com>

^{xiv} <http://en.wikinews.org/wiki>

^{xv} We do not attempt an abstractive technique in which novel sentences are generated. Abstraction would require an added sentence-building component – which would then affect the overall performance. Also, the influence of the pattern-generation component resp. abstraction component might be hard to disentangle.

^{xvi} This general statement is a conjecture, which we believe to be true due to the many degrees of freedom of how TTM algorithms may be defined. We invite attempts to prove or disprove this conjecture.

^{xvii} *maxMP* could also be used, but only adds new information when results are aggregated over periods and/or stories. In our experiment described in Section 5, we focused on the fine-grained results that only aggregate across users and therefore did not use *maxMP*.

^{xviii} <https://itunes.apple.com/sl/app/muse-summary-test/id643238319?mt=8>

^{xix} <http://narrativescience.com/>